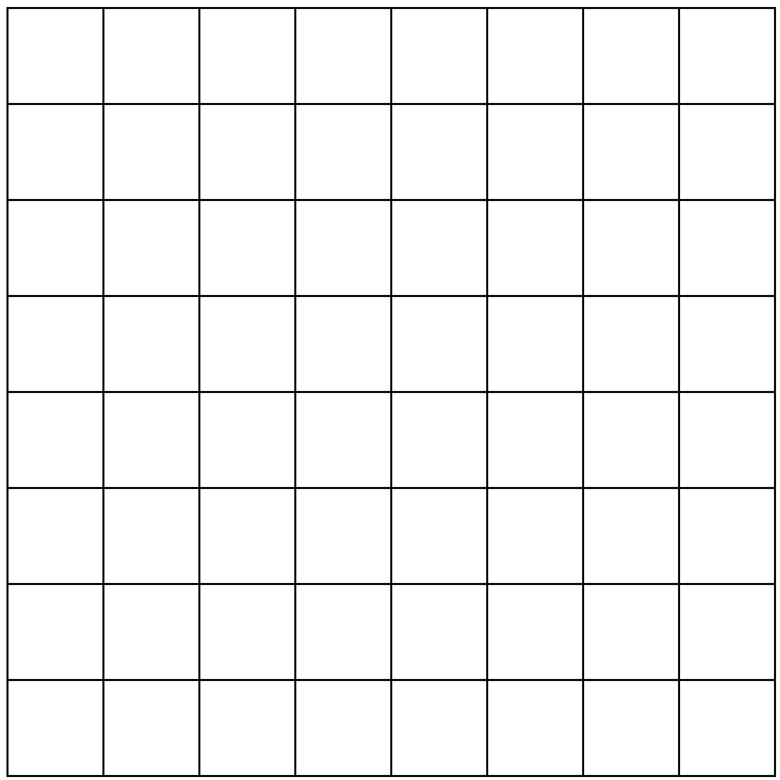
Introduction to Finite Automaton

- Consider the following game board that contains 64 boxes



Finite Automaton Continued …

There are some pieces of paper. Some are of white colour while others are of black color. The number of pieces of paper are 64 or less. The possible arrangements under which these pieces of paper can be placed in the boxes, are finite. To start the game, one of the arrangements is supposed to be initial arrangement. There is a pair of dice that can generate the numbers 2,3,4,…12 . For each number generated, a unique arrangement is associated among the possible arrangements.

Finite Automaton Continued …

It shows that the total number of transition rules of arrangement are finite. One and more arrangements can be supposed to be the winning arrangement. It can be observed that the winning of the game depends on the sequence in which the numbers are generated. This structure of game can be considered to be a finite automaton.

Defining Languages (continued)…

- Method 4 (Finite Automaton)
  Definition:
  A Finite automaton (FA), is a collection of the followings
  1) Finite number of states, having one initial and some (maybe none) final states.
  2) Finite set of input letters (Σ) from which input strings are formed.
  3) Finite set of transitions *i.e.* for each state and for each input letter there is a transition showing how to move from one state to another.

Example

- $\Sigma = \{a,b\}$
- **States:** x, y, z where x is an initial state and z is final state.
- **Transitions:**

1. At state **x** reading **a** go to state **z**,
2. At state **x** reading **b** go to state **y**,
3. At state **y** reading **a**, **b** go to state **y**
4. At state **z** reading **a**, **b** go to state **z**

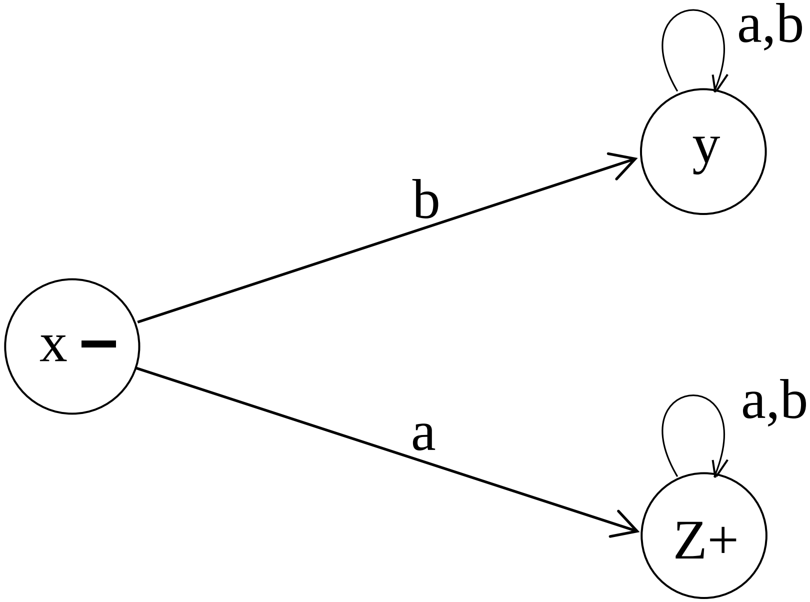
Example Continued …

- These transitions can be expressed by the following table called transition table

Old States	New States	
	Reading a	Reading b
x -	z	y
y	y	y
z +	z	z

Note

- It may be noted that the information of an FA, given in the previous table, can also be depicted by the following diagram, called the **transition diagram**, of the given FA



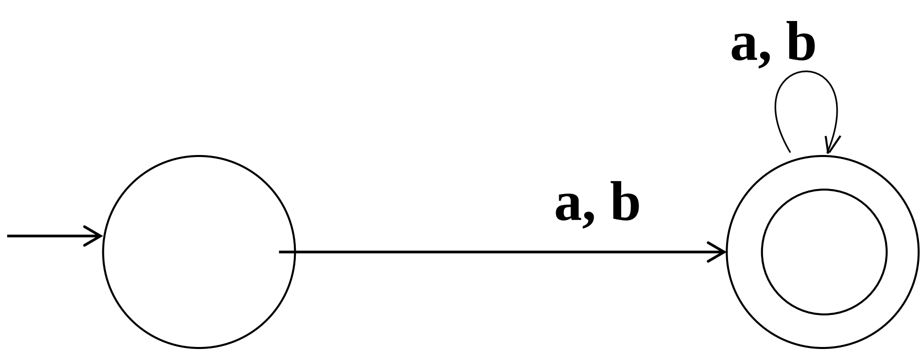
Remark

- The previous transition diagram is an FA accepting the language of strings, defined over $\Sigma=\{a, b\}$, **starting with a**. It may be noted that this language may be expressed by the regular expression

$$a\,(a + b)^*$$

Note

- It may be noted that to indicate the initial state, an arrow head can also be placed before that state and that the final state with double circle, as shown below. It is also to be noted that while expressing an FA by its transition diagram, the labels of states are not necessary.



Example

- $\Sigma = \{a,b\}$

  States: x, y, where x is both initial and final state.

  Transitions:

  1. At state x reading a or b go to state y.
  2. At state y reading a or b go to state x.

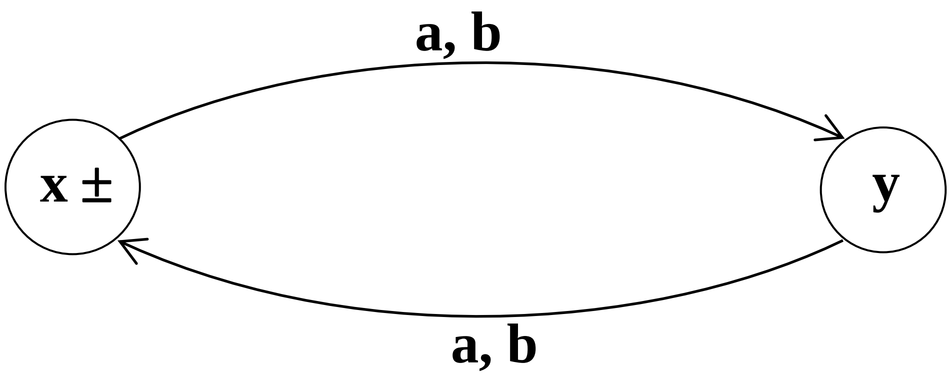
Example Continued …

- These transitions can be expressed by the following transition table

Old States	New States	
	Reading a	Reading b
x ±	y	y
y	x	x

Example Continued …

- It may be noted that the previous transition table may be depicted by the following transition diagram.



Example Continued …

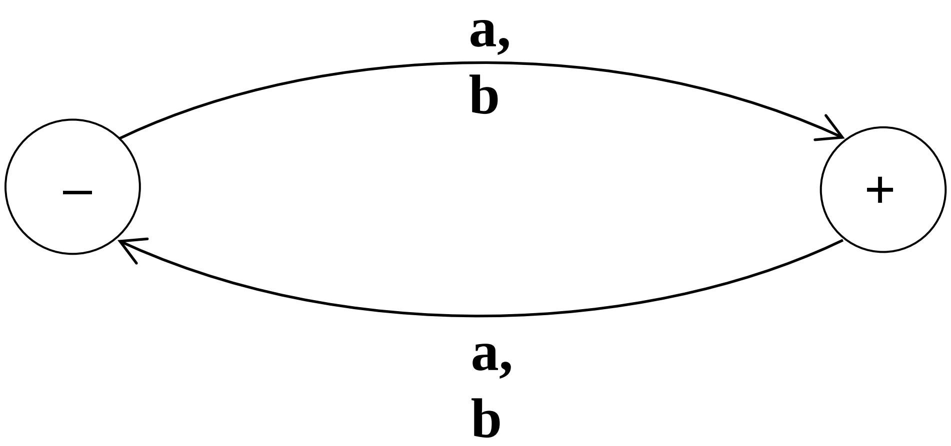
- The previous transition diagram is an FA accepting the language of strings, defined over $\Sigma=\{a, b\}$ of **even length**. It may be noted that this language may be expressed by the regular expression

$$((a+ b) (a + b))^*$$

TASK

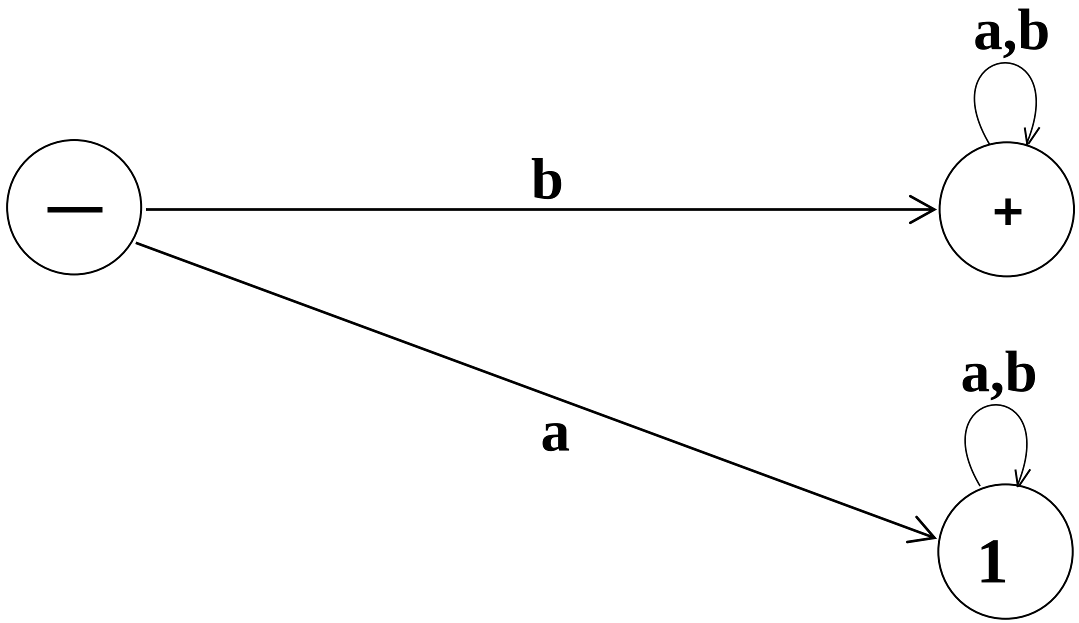
Build an FA for the language L of strings, defined over $\Sigma=\{a, b\}$, **of odd length.**

Solution of Task



Example: Consider the language L of strings, defined over $\Sigma=\{a, b\}$, **starting with b.** The language L may be expressed by RE $b(a + b)^*$

Solution

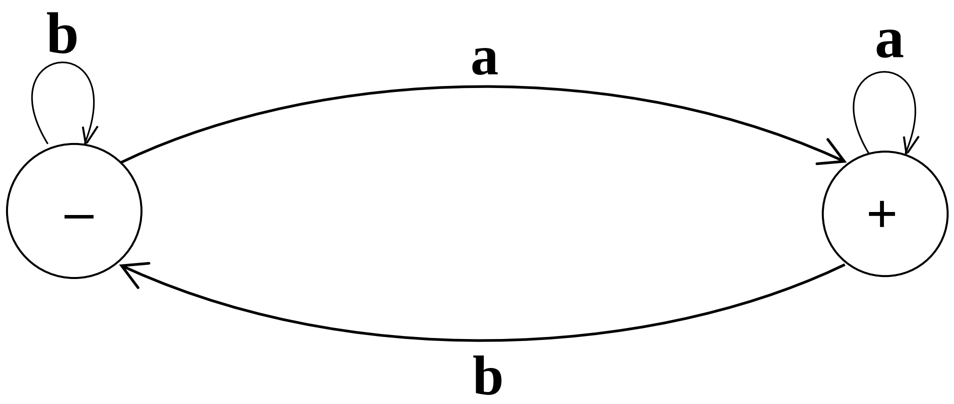


- **Example:**

  Consider the language L of strings, defined over $\Sigma=\{a, b\}$, **ending in a. T**he language L may be expressed by RE

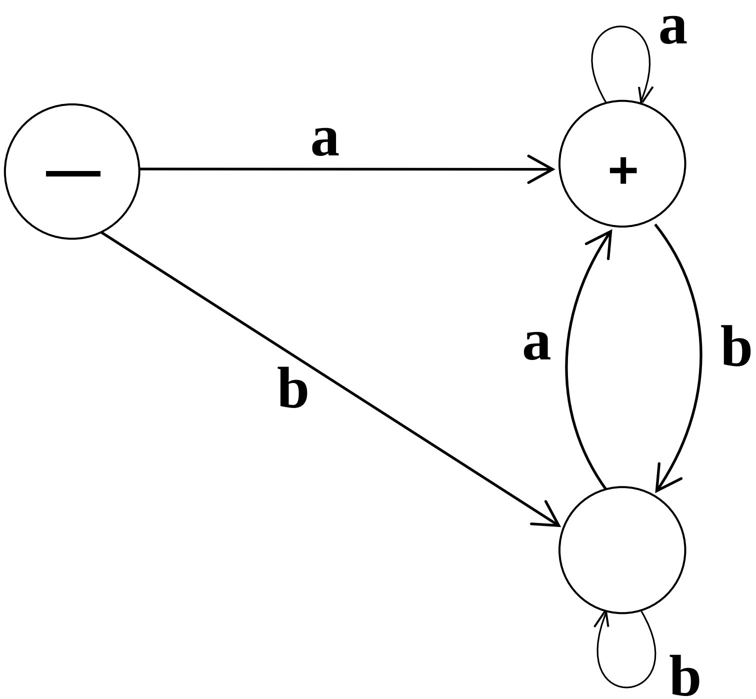
  $$(a+b)^*a$$

Example Continued …



There may be another FA corresponding to the given language.

Example continued ...



Remarks

- It may be noted that corresponding to a given language there may be more than one FA accepting that language, but for a given FA there is a unique language accepted by that FA.

Note

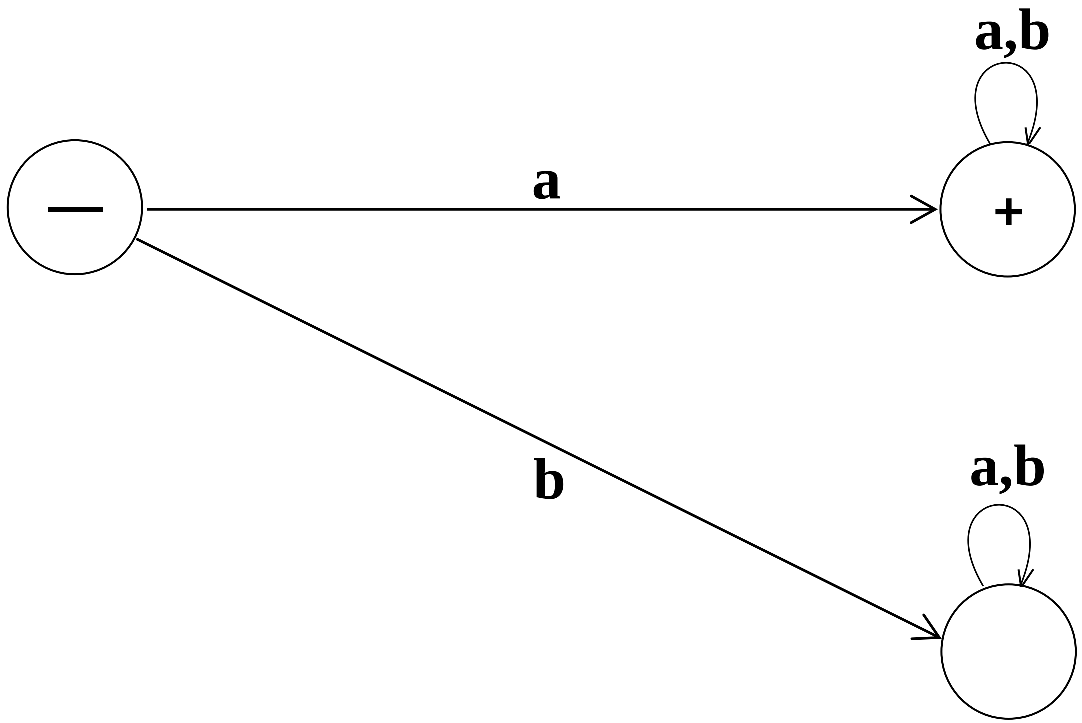
- It is to be noted that given the languages L_1 and L_2 ,where

L_1 = The language of strings, defined over $\Sigma=\{a, b\}$, **beginning with a**

L_2 = The language of strings, defined over $\Sigma=\{a, b\}$, **not beginning with b**

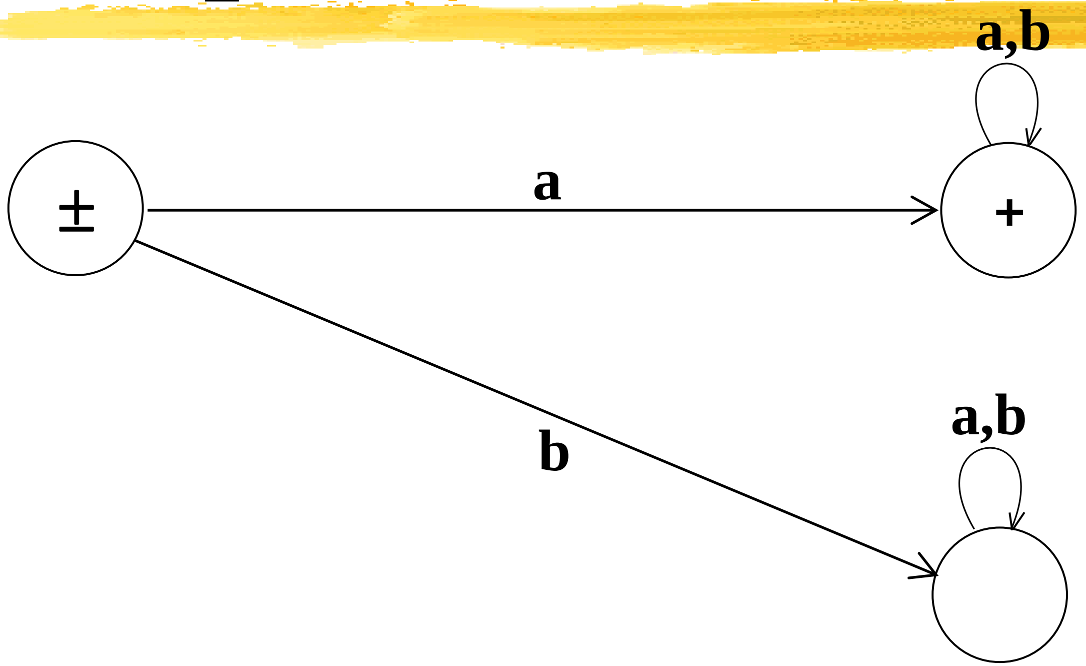
The Λ does not belong to L_1 while it does belong to L_2 . This fact may be depicted by the corresponding transition diagrams of L_1 and L_2.

FA_1 Corresponding to L_1



- The language L_1 may be expressed by the regular expression $a(a + b)^*$

FA$_2$ Corresponding to L$_2$



- The language L$_2$ may be expressed by the regular expression $a\,(a + b)^* + \Lambda$

Example

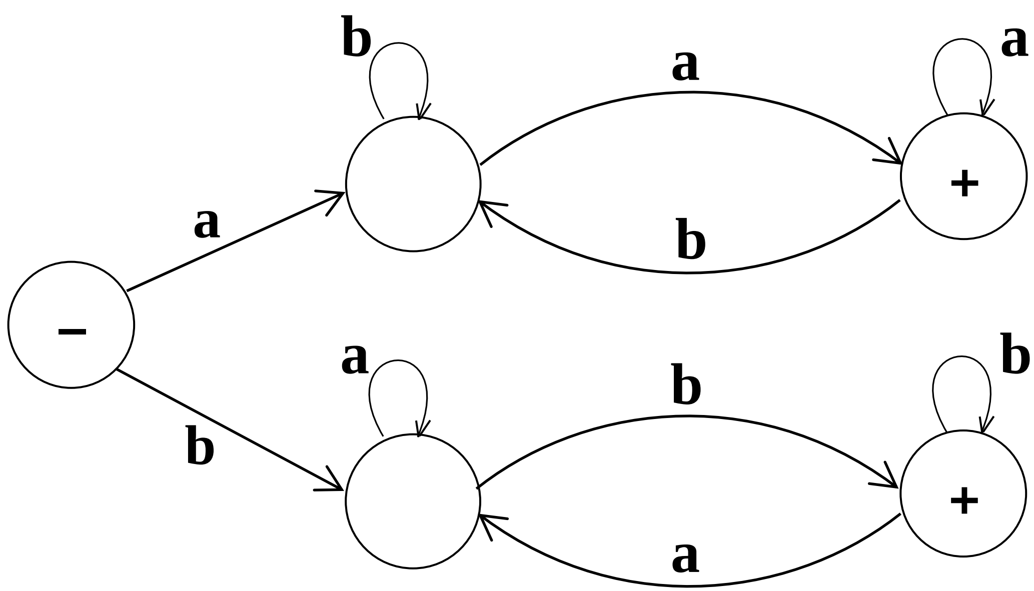
- Consider the Language L of Strings of **length two or more**, defined over Σ = {a, b}, **beginning with and ending in same letters.**

  The language L may be expressed by the following regular expression

  $$a\,(a + b)^{*}\,a + b\,(a + b)^{*}\,b$$

  It is to be noted that if the condition on the length of string is not imposed in the above language then **the strings a and b will then belong to the language.**

Example Continued ...



Task

- Build an FA accepting the Language L of Strings, defined over $\Sigma = \{a, b\}$, **beginning with and ending in same letters.**

Example

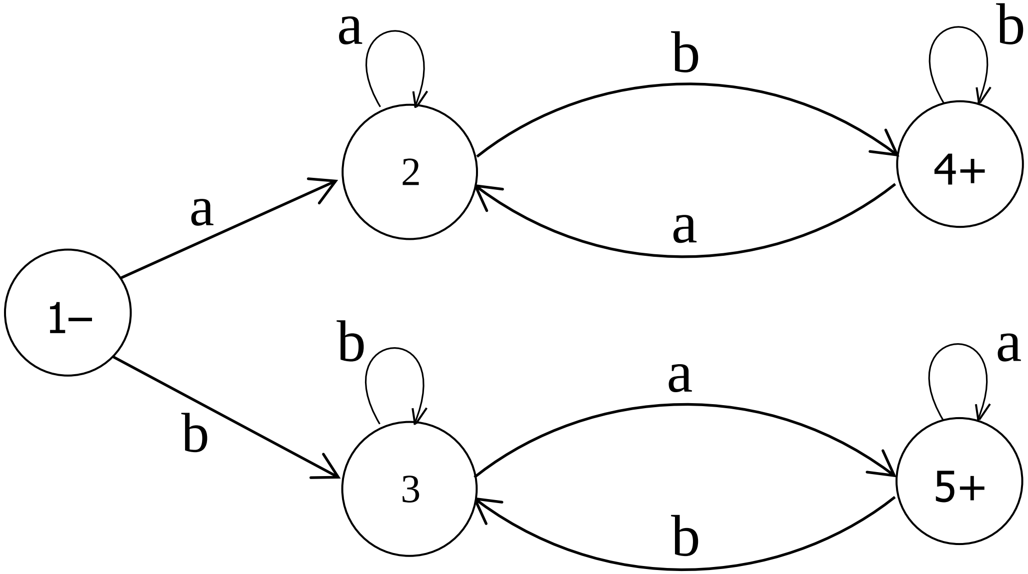
Consider the Language L of Strings , defined over $\Sigma = \{a, b\}$, **beginning with and ending in different letters.**

The language L may be expressed by the following regular expression

$$a\ (a + b)^*\ b + b\ (a + b)^*\ a$$

This language may be accepted by the following FA

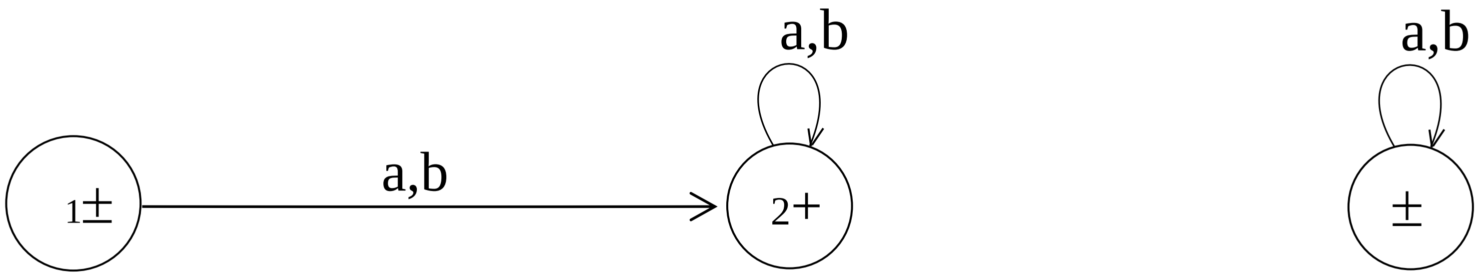
Example Continued ...



Example

- Consider the Language L , defined over $\Sigma = \{a, b\}$ of **all strings including Λ**

Example Continued …



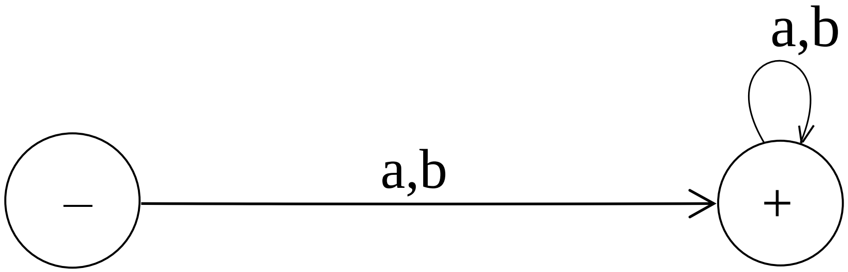
- The language L may be expressed by the following regular expression

$$(a + b)^*$$

Example

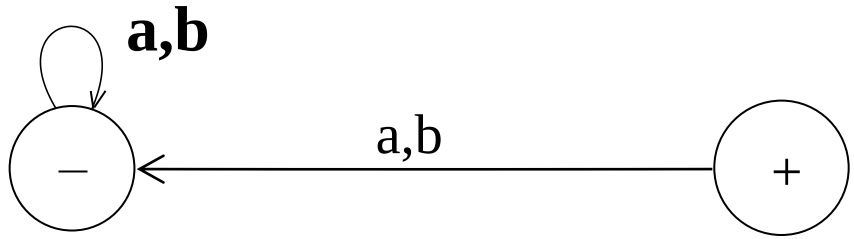
- Consider the Language L , defined over $\Sigma = \{a, b\}$ of **all non empty strings**.

Solution



Example

- Consider the following FA, defined over $\Sigma = \{a, b\}$



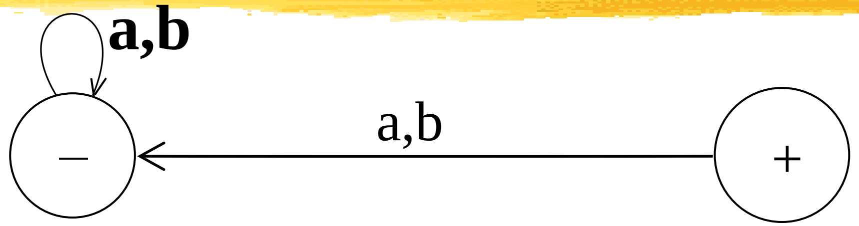
- It is to be noted that the above FA **does not accept any string**. Even it does not accept the null string. As there is no path starting from initial state and ending in final state.

Equivalent FAs

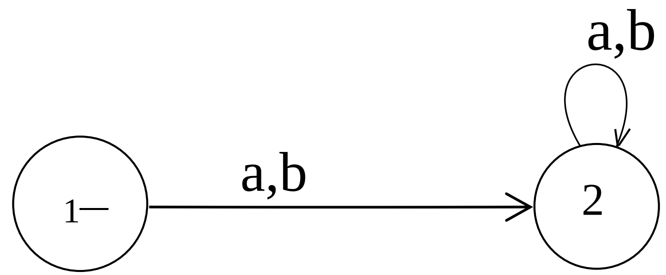
- It is to be noted that two FAs are said to be equivalent, if they accept the same language, as shown in the following FAs.

Equivalent FAs Continued …

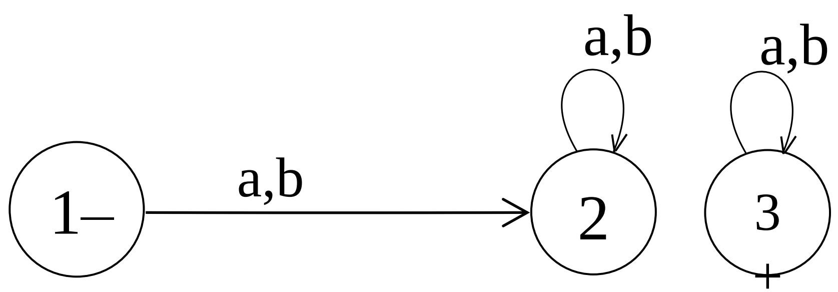
FA_1



FA_2



FA_3



Note (Equivalent FAs)

- FA_1 has already been discussed, while in FA_2, there is no final state and in FA_3, there is a final state but FA_3 is disconnected as the states 2 and 3 are disconnected.

  It may also be noted that the language of strings accepted by FA_1, FA_2 and FA_3 is denoted by the empty set *i.e.*

  { } OR Ø